

複雑系科学演習 レポート

公立はこだて未来大学 システム情報科学部 B3 日置竜輔

2021 年 10 月 26 日

目次

1	複雑系科学演習：レポート課題 1	2
2	複雑系科学演習：レポート課題 2	4
3	複雑系科学演習：レポート課題 3	6
4	複雑系科学演習：レポート課題 4	6

1 複雑系科学演習：レポート課題 1

課題 1

ロジスティック写像の時系列変化を計算するプログラムを作成し、 $r = 1.50, r = 2.60, r = 3.20, r = 3.50, r = 3.86, r = 3.90$ のとき、 $x_0 = 0.7$ として個体数変動の時系列グラフを表示せよ。

課題 2

ロジスティック写像のリターンマップを描くためのプログラムを作成し、 $r = 1.50, r = 2.60, r = 3.20, r = 3.50, r = 3.86, r = 3.90$ のとき、 $x_0 = 0.7$ として個体数変動のリターンマップを表示せよ。
グラフには、 $x_{n+1} = r(1 - x_n)x_n$ と $x_{n+1} = x_n$ のグラフも表示すること。

コードは以下のようになる。

Listing 1 Python によるロジスティック写像の描画

```
1 """
2 from matplotlib import pyplot as plt
3 from copy import copy
4 from collections import deque
5
6 class LogisticMap:
7     def __init__(self, start: float, constant: float, number: int, problem_number: int)
8         :
9         self.num = [i for i in range(number)]
10        tmp = start
11        self.n = deque([(tmp := constant * tmp * (1 - tmp)) for _ in self.num])
12        self.n.appendleft(start)
13        self.n.pop()
14        self.n = list(self.n)
15        self.problem_number = problem_number
16
17        x, y = copy(self.n), copy(self.n)
18        x.pop()
19        y.pop(0)
20        self.x = x
21        self.y = y
22
23    # 時系列グラフ
24    def time_series_graph(self):
25        fig = plt.figure()
26        plt.plot(self.num, self.n, marker="o", color="red", linestyle="--")
```

```

26     plt.xlabel("n",fontsize=14)
27     plt.ylabel("Xn",fontsize=14)
28     plt.xlim(0.0, max(self.num) * 1.1)
29     plt.ylim(0.0, max(self.n) * 1.1)
30     plt.savefig(f"/Users/ryusuke/Documents/git/FutureUniversityHakodate/
        ComplexScienceExercise/result/時系列グラフ
        week01_{self.problem_number}.png")
31     plt.close(fig)
32
33     # リターンマップ
34     def return_map(self):
35         fig = plt.figure()
36         plt.plot(self.x, self.y, marker="o", color="blue", linestyle="None")
37         plt.xlabel("Xn",fontsize=14)
38         plt.ylabel("Xn+1",fontsize=14)
39         plt.xlim(0.0, max(self.x) * 1.1)
40         plt.ylim(0.0, max(self.y) * 1.1)
41         plt.savefig(f"/Users/ryusuke/Documents/git/FutureUniversityHakodate/
        ComplexScienceExercise/result/リターンマップ
        week01_{self.problem_number}.png")
42         plt.close(fig)
43
44     l1 = LogisticMap(0.7, 1.5, 100, 1)
45     l1.return_map()
46     l1.time_series_graph()

```

2 複雑系科学演習：レポート課題 2

課題 1

ロジスティック写像で $r = 1.50, r = 2.60, r = 3.20, r = 3.50, r = 3.86, r = 3.90$ として、初期値 x_0 を 0 から 1 まで 0.001 きざみで変化させたときの、 x_{200} の値がどうなっているかグラフ化せよ。また、 x_n が $150 < n < 200$ の場合もグラフ化せよ。出力形式は授業資料を参照すること。

課題 2

課題 1 で得られた結果から初期値鋭敏性を説明せよ。

Listing 2 Python によるロジスティック写像の描画

```
1 import numpy as np
2 from matplotlib import pyplot as plt
3
4 class LogisticMap:
5     def __init__(self, constant: float, problem_number: int) -> None:
6         self.r = constant
7         self.x = np.linspace(0, 1, 1000)
8         self.problem_number = problem_number
9         fig = plt.figure(figsize=(12, 6))
10        plt.rcParams['font.family'] = 'AppleGothic'
11        self.ax1 = fig.add_subplot(1, 2, 1)
12        self.ax2 = fig.add_subplot(1, 2, 2)
13
14    def code_problem1(self) -> None:
15        ax1_array = []
16        for i in self.x:
17            num = i
18            for _ in range(200):
19                num = self.r * num * (1 - num)
20            ax1_array.append(num)
21        self.ax1.set_xlabel('$x_0$')
22        self.ax1.set_ylabel('$x_{200}$')
23        self.ax1.set_title(f'r={constant}, と  $x_0x_{200}$  の関係式')
24        self.ax1.plot(self.x, ax1_array, marker='.', linestyle='None')
25
26    def code_problem2(self) -> None:
27        ax2_array = []
28        x_array = []
29        for i in self.x:
30            num = i
31            for j in range(200):
```

```

32         num = self.r * num * (1 - num)
33         if 150 <= j:
34             ax2_array.append(num)
35             x_array.append(i)
36         self.ax2.set_xlabel('$x_0$')
37         self.ax2.set_ylabel('$x_{i}:(150 < i < 200)$')
38         self.ax2.set_title(f'r={constant}, X_150 ~ X_200 の関係式')
39         self.ax2.plot(x_array, ax2_array, marker='.', linestyle='None')
40
41     def show_graph(self) -> None:
42         plt.grid()
43         plt.savefig(f"/Users/ryusuke/Documents/git/FutureUniversityHakodate/
44             ComplexScienceExercise/result/week02_{self.problem_number}.png")
45
46
47     r = [1.50, 2.60, 3.20, 3.50, 3.86, 3.90]
48
49     for index, constant in enumerate(r):
50         Log = LogisticMap(constant, index)
51         Log.code_problem1()
52         Log.code_problem2()
53         Log.show_graph()

```

3 複雑系科学演習：レポート課題 3

課題 1

ロジスティック写像の初期変動の影響がないリターンマップを描くためのプログラムを作成し、個体数変動のリターンマップを表示せよ。

このとき、 r は、 $r = 1.50$, $r = 2.60$, $r = 3.20$, $r = 3.50$, $r = 3.86$, $r = 3.90$ のとし、初期値 x_0 はランダムに与えなさい。

グラフは、授業資料を参考として、 $x_{n+1} = r(1 - x_n)x_n$, $x_{n+1} = x_n$ も同時に描画し、縦軸と横軸は 0~1 の範囲で出力すること。

課題 2

r が 1~4 のときのロジスティック写像の分岐図を描け。また、分岐図の中で 3 周期の窓が現れている r の範囲を抽出して、グラフを描け。

このとき、両グラフとも r は各自適切な刻み幅を設定し、各 r について初期値 x_0 をランダムに与えること。プログラムのソースコードは、 r が 1~4 のときの分岐図を出力するものと 3 周期の窓を出力するものの 2 つを記載すること。

課題 3

課題 1 と課題 2 は、各 r ごとに初期値 x_0 をランダムに与えているにもかかわらず、 r が 1~3.5 くらいまでは何度プログラムを実行しても同じようなグラフを描くことができる。

一方、 r が 3.5 よりも大きくなっていくと、プログラムを実行するたびにグラフを一見するだけではわからないような違いが生じる。

この理由を前回の課題と初期値鋭敏性という言葉とを用いて説明せよ。

4 複雑系科学演習：レポート課題 4

課題 1

リアプノフ指数の r 依存性を示したグラフを描け。

但し、初期値をランダムに与え、グラフの横軸は 1~4、縦軸は -3~1 までの範囲にすること。

r の刻み幅は、各自適切な値を設定すること。

課題 2

3 周期の窓の領域でのリアプノフ指数の r 依存性を示したグラフを描け。
但し、初期値をランダムに与え、グラフの縦軸は $-1 \sim 0.4$ までの範囲にすること。
 r の範囲および刻み幅は、各自適切な値を設定すること。

課題 3

ロジスティック写像についてまとめ、これまでに出题された全て (4 回分) の結果について考察せよ。
分量は A4 用紙 1~4 枚程度を目安としてください。

その他

4 回分の講義についての感想・意見等があれば、書いてください。(採点対象ではありません。)